Transparência de Domínios: maior auditabilidade para serviços de Transparência de Certificados

Fernando K. Matsumoto[1], **Jonatas F. Viana da Silva**[1], **Marcos A. Simplicio Junior**[1]

[1]Escola Politécnica
Universidade de São Paulo (USP) – São Paulo, SP — Brazil

{fer.matsumoto,jonatas_ferreira}@usp.br, mjunior@larc.usp.br

Abstract. *Web authentication relies critically on the Certificate Authorities (CAs), so it is essential to verify that they are operating correctly. In this context, the Certificate Transparency (CT) initiative was created with the objective of facilitating the verification of CAs. In particular, using CT, it is possible to identify possible fraudulent certificates issued by CAs. However, the services currently available to perform this identification have been shown to be unreliable, returning sometimes incomplete results. This article presents a tool implementing the concept of Domain Transparency (DT), which uses verifiable data structures to offer a reliable and trustworthy alternative to these services.*

Resumo. *A autenticação na Internet depende criticamente das Autoridades Certificadoras (Certificate Authorities – CAs), de forma que é essencial verificar a operação correta das CAs. Nesse contexto, a iniciativa conhecida como Transparência de Certificados (Certificate Transparency – CT) foi criada com o objetivo de facilitar a verificação das CAs. Em particular, utilizando CT, é possível identificar eventuais certificados fraudulentos emitidos pelas CAs. No entanto, os serviços disponíveis atualmente para realizar essa identificação já se mostraram pouco confiáveis, retornando por vezes resultados incompletos. Este artigo apresenta uma ferramenta que implementa o conceito de Transparência de Domínios (Domain Transparency – DT), utilizando estruturas de dados verificáveis para oferecer uma alternativa confiável a esses serviços.*

1. Introdução

Tradicionalmente, a autenticação web é feita pela integração de TLS com certificados X.509 [Rescorla 2018], permitindo que os usuários verifiquem a identidade do servidor que pretendem acessar. Esses certificados são comumente emitidos usando uma Infraestrutura de Chaves Públicas (*Public Key Infrastructure* – PKI), i.e., certificados são assinados por Autoridades Certificadoras (*Certificate Authorities* – CAs) Intermediárias, cujos certificados são por sua vez assinados pelas CAs raiz [Boeyen et al. 2008]. Os sistemas operacionais e navegadores web são então pré-carregados com uma série de CAs raízes, o que permite que eles aceitem todos certificados digitais emitidos por essas CAs.

Embora essa abordagem seja funcional e amplamente utilizada, ela também carrega um risco: como tal delegação de confiança depende de um conjunto de CAs, a PKI depende criticamente da operação correta e da confiabilidade dessas autoridades. Portanto, caso uma CA emita certificados não autorizados, a confiabilidade do sistema fica fortemente comprometida. Isto aconteceu, por exemplo, em 2011, quando a CA

DigiNotar foi invadida e o intruso obteve um certificado fraudulento para o domínio `google.com`. Este certificado permitiu a realização de um ataque de personificação do Google no Irã, atingindo cerca de 300 000 usuários [Hoogstraaten et al. 2012].

Na tentativa de contrapor esse risco, uma dificuldade é identificar certificados fraudulentos, visto que, tradicionalmente, não é possível listar os certificados que foram emitidos por CAs. A iniciativa conhecida como Transparência de Certificados (*Certificate Transparency* – CT) busca resolver esse problema, aumentando a transparência das CAs [Laurie et al. 2013]. O sistema cria uma série de logs públicos nos quais todos os certificados emitidos por CAs devem ser cadastrados. Esses logs permitem que um usuário enumere todos os certificados emitidos pelas CAs, permitindo o seu monitoramento e auditoria. Nesse sentido, a existência de CTs em operação dificulta a emissão de certificados fraudulentos sem que o proprietário do domínio tome conhecimento do ocorrido. É importante salientar, entretanto, que CTs não impedem nem têm o papel de detectar a emissão de certificados fraudulentos. Sua finalidade é essencialmente disponibilizar certificados em um registro acessível publicamente, para que terceiros interessados possam acessá-los e verificá-los. Exatamente por isso, seria esperado que toda CT oferecesse uma maneira fácil de filtrar certificados por domínio, permitindo que o titular de cada domínio identifique certificados fraudulentos sem processar os logs completos (um volume estimado em 28 GB de dados por dia em 2018 [Li et al. 2019]). Infelizmente, embora tais serviços existam, eles não são totalmente confiáveis, pois vários certificados registrados em CTs não são retornados pelos serviços de busca que deveriam fazê-lo [Li et al. 2019].

Com o objetivo de trazer ainda mais transparência e utilidade para CTs, este artigo propõe o conceito de Transparência de Domínios (*Domain Transparency* – DT), uma alternativa aos serviços de pesquisa existentes atualmente. Essa alternativa opera com garantias criptográficas de confiabilidade, permitindo a enumeração confiável de certificados para um domínio específico. A ferramenta aqui apresentada[1] implementa esse conceito de modo compatível com a arquitetura de CTs atual, apresentando-se como uma extensão (não um substituto) aos mecanismos já disponíveis em servidores de CT em operação.

O resto deste artigo é assim organizado. A Seção 2 discute brevemente as tecnologias utilizadas em Transparência de Domínios. A Seção 3 apresenta o sistema proposto. Por fim, a Seção 4 conclui a discussão e apresenta direcionamentos para trabalhos futuros.

2. Background

Esta seção discute os principais conceitos que compõem a presente proposta: estruturas de dados verificáveis e a iniciativa conhecida como Transparência de Certificados.

2.1. Estruturas de Dados Verificáveis

Um aspecto importante em Transparência de Domínios é a verificabilidade: os clientes devem ser capazes de verificar as respostas retornadas pela Transparência de Domínios. Para isso, serão utilizadas estruturas de dados verificáveis, conforme proposto em [Eijdenberg et al. 2015]. As estruturas propostas pelos autores serão descritas abaixo.

[1] Repositório de DT — `https://github.com/fernandokm/transparencia-de-dominios`

Logs Verificáveis Um log verificável é uma lista *append-only* (i.e. uma lista na qual elementos podem ser adicionados, mas não removidos), como a usada em Transparência de Certificados [Laurie et al. 2013] e em Blockchains [Nakamoto 2008, Swan 2015]. Periodicamente, um log verificável publica uma *cabeça de árvore assinada* (*Signed Tree Head* – STH), que é uma estrutura de dados acompanhada de assinatura digital. Utilizando essa cabeça, o cliente pode:

1. dado um elemento, obter uma prova criptográfica de que ele realmente estava presente no log no momento em que a cabeça foi gerada (*prova de auditoria*); e
2. dada uma outra cabeça, gerada previamente, verificar que o log é *append-only*, ou seja, que todos os elementos que estavam presentes na cabeça antiga ainda estão presentes na cabeça atual (*prova de consistência*).

O item 1 significa que clientes distintos podem comparar apenas a cabeça e ter a certeza de que todos estão vendo os mesmos elementos. Além disso, qualquer discrepância observada pode ser imediatamente reportada. Como a cabeça é assinada pelo log, ela não pode ser forjada por atacantes, de forma que qualquer erro presente nos elementos do log pode ser atribuído, pelo menos em parte, a algum erro ou comportamento fraudulento do log.

Mapas Verificáveis Um mapa verificável é um mapa de chaves arbitrárias a valores arbitrários, como o usado em Transparência de Revogação [Laurie and Cutter 2012]. De forma análoga aos logs, um mapa gera, periodicamente, uma *cabeça de mapa assinada* (*Signed Map Head* – SMH), que é acompanhada de uma assinatura digital. Dada uma cabeça de mapa, é possível então produzir provas de pertencimento e não pertencimento, que provam, respectivamente, que uma chave k tem valor v ou que essa chave não tem nenhum valor. Assim como ocorre com os logs, a cabeça de mapa permite que os clientes detectem comportamentos inválidos do mapa (i.e. fornecer resultados diferentes dependendo do cliente) e provem que esse comportamento inválido realmente ocorreu. De forma contrária ao log verificável, que tinha provas de consistência, não é possível validar o comportamento de um mapa verificável ao longo do tempo.

Mapas verificáveis apoiados em log Para possibilitar a validação do comportamento do mapa no tempo, é necessário combinar o mapa verificável com um log verificável. Nesse caso, o log verificável serve como um histórico de todos os eventos que levaram ao estado atual do mapa (e.g., cada elemento do log pode indicar que uma chave foi alterada), permitindo que clientes verifiquem como os valores do mapa estão sendo modificados.

2.2. Transparência de Certificados

Transparência de Certificados (*Certificate Transparency* – CT) é uma iniciativa que utiliza logs verificáveis para rastrear todos os certificados emitidos pelas CAs [Laurie et al. 2013]. Além dos logs, CT envolve duas outras entidades:

- os *monitores*, que verificam os logs de CT, buscando certificados suspeitos e comportamentos inválidos dos logs (e.g. remoção de certificado); e
- os *auditores*, que verificam se os logs estão incluindo todos os certificados que eles prometeram incluir — e.g. um dono de domínio pode submeter os seus certificados para um log e verificar que todos eles foram devidamente incluídos.

Conforme descrito na Seção 2.1, o uso dos logs verificáveis significa que os clientes que utilizam CT podem confiar nos dados, visto que eles já foram validados pelos monitores e auditores.

Esse sistema, a priori, não tem nenhuma forma de garantir que todos os certificados sejam inclusos nos logs. Para resolver esse problema, a maioria dos clientes e navegadores web, como o Google Chrome [Google LLC 2021], aceitam apenas certificados que estejam presentes em algum log.

Conforme foi mencionado na introdução, a busca por certificados fraudulentos pode ser excessivamente cara para os proprietários de domínio. A solução para esse problema é oferecida por alguns monitores que, além de verificarem o comportamento correto dos logs, também fornecem serviços de busca de certificados por domínio. Para explicar esse comportamento por parte dos monitores, basta observar que eles já processam todos os certificados em busca de inconsistências, o que facilita fortemente o fornecimento de tal serviço. No entanto, sabe-se que esses serviços não são completamente confiáveis, retornando resultados por vezes incompletos [Li et al. 2019].

3. Transparência de Domínios

Conforme discutido em [Li et al. 2019], os serviços de busca oferecidos atualmente pelos monitores em soluções de CT apresentam confiabilidade insatisfatória. Para resolver os problemas descritos, Transparência de Domínios (*Domain Transparency* – DT) fornece um serviço de busca *verificável*. As funcionalidades de DT são descritas em mais detalhe a seguir.

Rastreamento automático de logs Para fornecer o serviço de busca, cada servidor de DT mantém uma lista de logs de Transparência de Certificados, chamados de *logs fonte*, que são periodicamente verificados em busca de novos certificados. A lista de logs fonte não é fixa: novos logs podem ser adicionados quando necessário. Isso permite que logs novos sejam incorporados sem a necessidade de se resetar o sistema. Como será discutido na Seção 3.5, essa característica é particularmente interessante em vista do esquema de *sharding* utilizado atualmente em Transparência de Certificados.

Busca de certificados Dado um domínio, clientes podem obter do servidor de DT uma lista de certificados associados àquele domínio. Essa lista contém *todos* os certificados que estão presentes nos logs fonte e que estão associados ao domínio especificado. Além disso, é possível baixar essa lista de forma incremental (somente a parte nova da lista precisa ser baixada pelo cliente).

Sistema verificável De forma análoga ao que é feito em Transparência de Certificados, é possível verificar se o servidor de DT está se comportando de forma correta. Para possibilitar a verificação dos resultados, o servidor periodicamente emite uma assinatura digital que serve como uma promessa de que ele está operando corretamente. Os mesmos *monitores* que operam em CT podem então verificar que essa promessa está sendo mantida. Por fim, donos de domínio podem obter a lista de certificados associados ao seu domínio e validá-la utilizando a assinatura produzida pelo servidor de DT. Da mesma forma que

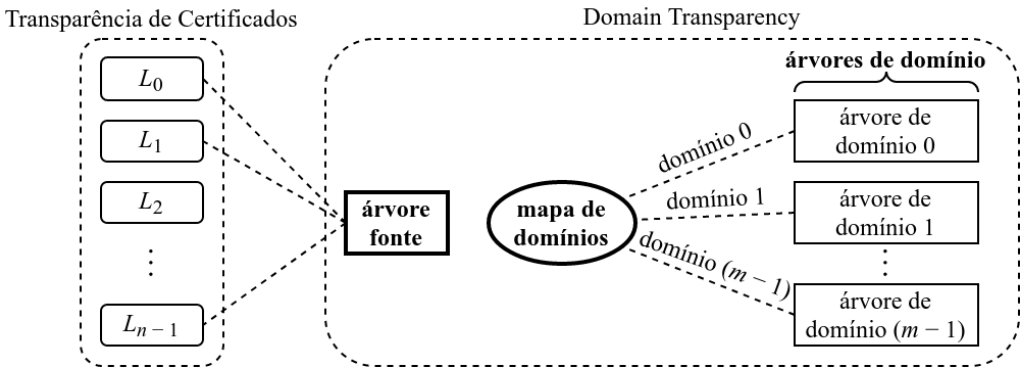

Figura 1. Visão geral de Transparência de Domínios. O mapa de domínios associa cada domínio a uma árvore de domínios e a árvore fonte lista os logs fonte. Cada árvore de domínio armazena, dentre todos os certificados presentes nos logs fonte, aqueles que estão associados ao seu domínio.

ocorre em CT, esse sistema utiliza assinaturas digitais para garantir que comportamentos maliciosos por parte dos servidores de DT possam ser rapidamente detectados e atribuídos de forma inequívoca aos servidores.

3.1. Arquitetura

Para implementar as funcionalidades descritas na seção anterior, DT utiliza estruturas de dados verificáveis. De forma simplificada, um servidor de DT mantém, para cada domínio, uma lista *append-only* com todos os certificados daquele domínio que aparecem nos logs fonte. Mais especificamente, DT tem 3 componentes principais: um log fonte, um mapa de domínios e um conjunto de árvores de domínio, conforme pode ser visto na Figura 1.

Primeiramente, é necessário definir quais logs CT serão rastreados pelo sistema (os *logs fonte*). Para isso, é utilizado um log verificável chamado *árvore fonte*, que guarda os IDs de todos os logs que são rastreados. Ressalta-se que é possível adicionar elementos na árvore fonte, o que é importante para lidar com logs de CT novos. Em seguida, define-se, um *mapa de domínios*, que é um mapa verificável que associa cada domínio a um log verificável chamado de *árvore de domínio*. Essa árvore de domínio contém uma lista de tuplas (i, j), indicando que o certificado na j-ésima posição do i-ésimo log fonte de CT está associado ao respectivo domínio. Ressalta-se que, caso um certificado esteja presente em vários logs, haverá várias tuplas (i, j) correspondentes a esse certificado (uma para cada log). Nesse caso, todas as tuplas são incluídas na árvore de domínio.

3.2. Mapa de Domínios

Conforme especificado acima, o mapa de domínios é um mapa verificável que mapeia cada domínio a uma árvore de domínio. Sempre que o mapa é atualizado (i.e. novos certificados são inseridos nas árvores de domínio), é gerada uma nova *cabeça do mapa*, que é uma estrutura de dados contendo os seguintes atributos:

- a versão do sistema (atualmente v1);
- um timestamp;
- a raiz do mapa e o seu tamanho (a soma do tamanho de suas árvores);
- a raiz da árvore fonte; e

```
sh-5.1$ curl http://127.0.0.1:6962/dt/v1/get-smh
{"timestamp":1626732913,"map_size":43,"map_root_hash":"zSr4aIy/jFAlZjBAEFr8vCzIxuPiGv9GlbyrgD+kXxs=","source_tree_root_hash":"AtDTMdBbCJ
JmehTxF2GfXQLjExzBH0SAR6jpKVyktl0=","source_log_revisions":[{"tree_size":43,"root_hash":"VYuT4xXn0BLhoLdTUObXucMPrlAJZm5W90bNOFgB42c="}]
,"map_head_signature":"MEUCIH3Gcq3PjST7rhKs8/4Jsz0yOaDcOBSqVPJk9bH3rhFTAiEAtn1a9j3TWmY3ZwxWNk4Wq/G9DxMp5IvJXeVoZMgfMjM="}
```

Figura 2. Exemplo de uso da API de DT para obter o SMH mais recente.

- as raízes e tamanhos de cada um dos logs fonte.

Ressalta-se que esses atributos são suficientes para especificar quais são os logs fonte e quais as últimas revisões de cada log fonte presentes no mapa. Essa cabeça é então assinada, produzindo uma cabeça de mapa assinada (SMH). O SMH serve como uma promessa de que todos os certificados presentes nos logs fonte (nas raízes especificadas) estão presentes no mapa.

Monitores podem então verificar se o mapa de DT está operando corretamente. Em caso negativo, pode-se utilizar a assinatura digital armazenada no SMH para associar, de forma inequívoca, o comportamento errôneo ao servidor de DT.

Para garantir que certificados sejam incluídos no mapa em tempo hábil, cada mapa tem um atraso máximo de mesclagem (*Maximum Merge Delay* – MMD), que é o tempo máximo após um certificado ser incluído num log CT até que ele apareça no mapa.

3.3. Provas Criptográficas

Como DT é composto por estruturas de dados verificáveis, todas as provas descritas na Seção 2.1 são suportadas. Em particular as seguintes provas são úteis:

- prova de não pertencimento do mapa de domínios (prova de que não existe árvore de domínio para algum domínio específico, ou seja, de que não existem certificados para esse domínio nos logs fonte);
- prova de pertencimento da árvore fonte; e
- prova de consistência da árvore fonte.

Além disso, em alguns casos, é necessário combinar mais de uma prova:

- para provar que um certificado está associado a um domínio d, é necessário provar que o certificado está presente numa árvore de domínio (prova de pertencimento da árvore de domínio) e que essa árvore de domínio está associada a d no mapa de domínios (prova de pertencimento do mapa de domínios); e
- analogamente, para provar que uma árvore de domínio é *append-only*, é necessário combinar provas de pertencimento do mapa de domínios com uma prova de consistência da árvore de domínio.

3.4. Protocolo e Interação com a Ferramenta

A interação com um servidor de DT é feita utilizando uma API HTTP/HTTPS similar à utilizada em CT, como mostrado na Figura 2. Para interagir com cada um dos componentes de DT, o usuário realiza requisições HTTP de método GET, passando os parâmetros por meio da *query string*. A resposta é dada no formato JSON. A Figura 3 mostra um exemplo de um programa construído utilizando esses comandos.

Para o mapa de domínios, clientes podem:

- obter a cabeça assinada (SMH) mais recente; e

```
2021/07/10 19:07:00 Domain tracker started...
2021/07/10 19:07:14 New SMH: timestamp=1626732433, size=41, rootHash=eedbed0a..., sourceRootHash=02d0d331..., sourceLogCount=1
2021/07/10 19:07:14 New certificate for example.com:
  SHA-256 Fingerprint: 200DCAFA767C8450ECE644879C062A0CDF52240FE05BB7EB284611C3AEF3EC2E
  Leaf Index: 40
```

Figura 3. Exemplo de aplicação de DT. Essa aplicação utiliza DT para notificar o usuário sempre que um certificado novo é adicionado nos logs fonte.

- dado um domínio, obter a raiz da respectiva árvore de domínio, assim como uma prova de pertencimento.

Dada uma árvore de domínio obtida com os comandos acima, um cliente pode:

- verificar a consistência entre duas revisões (raízes) da árvore;
- obter todos os certificados num dado intervalo (implementações podem impor um número máximo de certificados por pedido);
- obter o certificado num índice específico, assim como uma prova de auditoria; e
- obter a posição de um par (certificado, log fonte) na árvore, caso esse par esteja presente.

Por fim, clientes também podem interagir com a árvore fonte:

- verificar a consistência entre duas revisões (raízes) da árvore fonte;
- obter todos os logs fonte num dado intervalo (implementações podem impor um número máximo de logs por pedido); e
- obter o log fonte num índice específico, assim como uma prova de auditoria.

3.5. Detalhes Operacionais

Os mapas de DT podem ser operados por qualquer entidade interessada. Contudo, existem algumas entidade mais propícias a essa atividade. Destaca-se os operadores de logs e os monitores já existentes no ecossistema de CT. Ambas essas entidades já possuem fácil acesso a um conjuntos de logs de CT e conseguem verificar quando esses logs são atualizados.

Cada monitor de CT já rastreia atualizações de um conjunto de logs de CT. Esse mesmo conjunto pode ser utilizado como o conjunto de logs fonte do mapa, de forma que não seja necessária nenhuma comunicação adicional com os logs fonte. Nesse caso, como o monitor precisa obter todos os certificados dos logs, é justificável um MMD grande, como 24 horas. Já os operadores de log de CT possuem em sua disposição apenas os seus logs. Logo, eles podem decidir executar um mapa que use os seus logs como logs fonte. Nesse caso, o MMD deve ser menor, visto que eles tem acesso direto e sabem quando os logs são atualizados.

Um último detalhe operacional se refere ao mecanismo comumente utilizado por operadores de CT para evitar o crescimento ilimitado dos logs de CT, conhecido como fragmentação temporal ou *sharding* [Lynch 2018]: Nesse esquema, cada log deve ser fragmentado em vários logs *lógicos*, de forma que cada log lógico armazene apenas certificados que expiram em um ano específico. Por exemplo, a DigiCert opera os logs lógicos Yeti 2021 e Yeti 2022, que armazenam, respectivamente, certificados com data de expiração em 2021 e 2022. Essa mesma estratégia pode ser adotada por operadores de DT: cada mapa de DT pode utilizar como logs fonte apenas os logs lógicos relativos a um determinado ano. Da mesma forma como ocorre em CT, essa técnica ajuda a limitar o tamanho dos mapas, assim como o número de logs fonte de cada mapa.

4. Conclusão e Trabalhos Futuros

Transparência de Domínios é uma ferramenta que estende a infraestrutura já existente de CT, permitindo que clientes encontrem certificados fraudulentos de forma fácil e confiável. Além disso, o sistema é facilmente integrável na arquitetura já existente: os mapas de DT podem ser operados pelos monitores e operadores de log já existentes.

Para resolver os problemas existentes nos serviços de busca oferecidos pelos monitores, foram utilizadas estruturas de dados verificáveis semelhantes às utilizadas atualmente por Transparência de Certificados.

O mapa apresentado neste artigo armazena apenas os certificados associados a cada domínio. Em [Laurie and Cutter 2012], os autores apresentam uma proposta inicial de Transparência de Revogação, ou seja, a possibilidade de armazenar informações sobre quais certificados foram revogados. Como trabalho futuro, planeja-se estender as ideias de Transparência de Revogação, incorporando-as em DT.

Um outro ponto de trabalho futuro é a melhoria da experiência de usuário da ferramenta, incluindo a criação de uma interface gráfica (aplicação web).

Agradecimentos: Este trabalho foi financiado pela Ripple via *University Blockchain Research Initiative* (UBRI) e pelo CNPq (bolsa de produtividade 304643/2020-3).

Referências

Boeyen, S., Santesson, S., Polk, T., Housley, R., Farrell, S., and Cooper, D. (2008). Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280.

Eijdenberg, A., Laurie, B., and Cutter, A. (2015). Verifiable data structures. White paper, Google LLC.

Google LLC (2021). Chrome certificate transparency policy.

Hoogstraaten, H., Prins, R., Niggebrugge, D., Heppener, D., Groenewegen, F., Wettinck, J., Strooy, K., Arends, P., Pols, P., Kouprie, R., Moorrees, S., van Pelt, X., and Hu, Y. Z. (2012). Black tulip: Report of the investigation into the diginotar certificate authority breach. Technical report, Fox-IT.

Laurie, B. and Cutter, A. (2012). Revocation transparency. White paper, Google LLC.

Laurie, B., Langley, A., and Kasper, E. (2013). Certificate Transparency. RFC 6962.

Li, B., Lin, J., Li, F., Wang, Q., Li, Q., Jing, J., and Wang, C. (2019). Certificate transparency in the wild: Exploring the reliability of monitors. In *Proc. of the ACM SIGSAC Conference on Computer and Communications Security*, pages 2505–2520.

Lynch, V. (2018). Scaling CT logs: Temporal sharding. `www.digicert.com/blog/scaling-certificate-transparency-logs-temporal-sharding/`.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260.

Rescorla, E. (2018). The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446.

Swan, M. (2015). *Blockchain: Blueprint for a new economy*. O'Reilly Media, Inc.